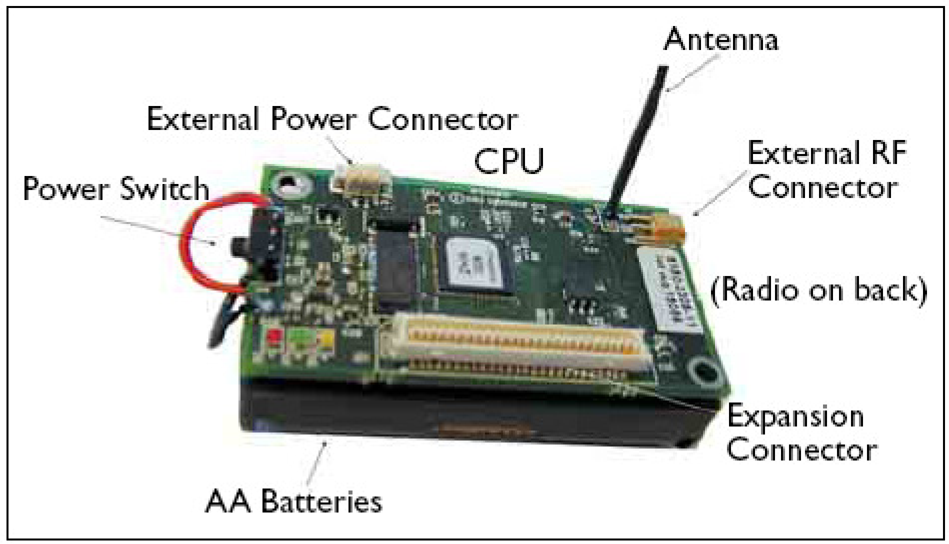
How can Computational Geometry help Mobile Networks:

Geometry in Wireless Sensor Networks

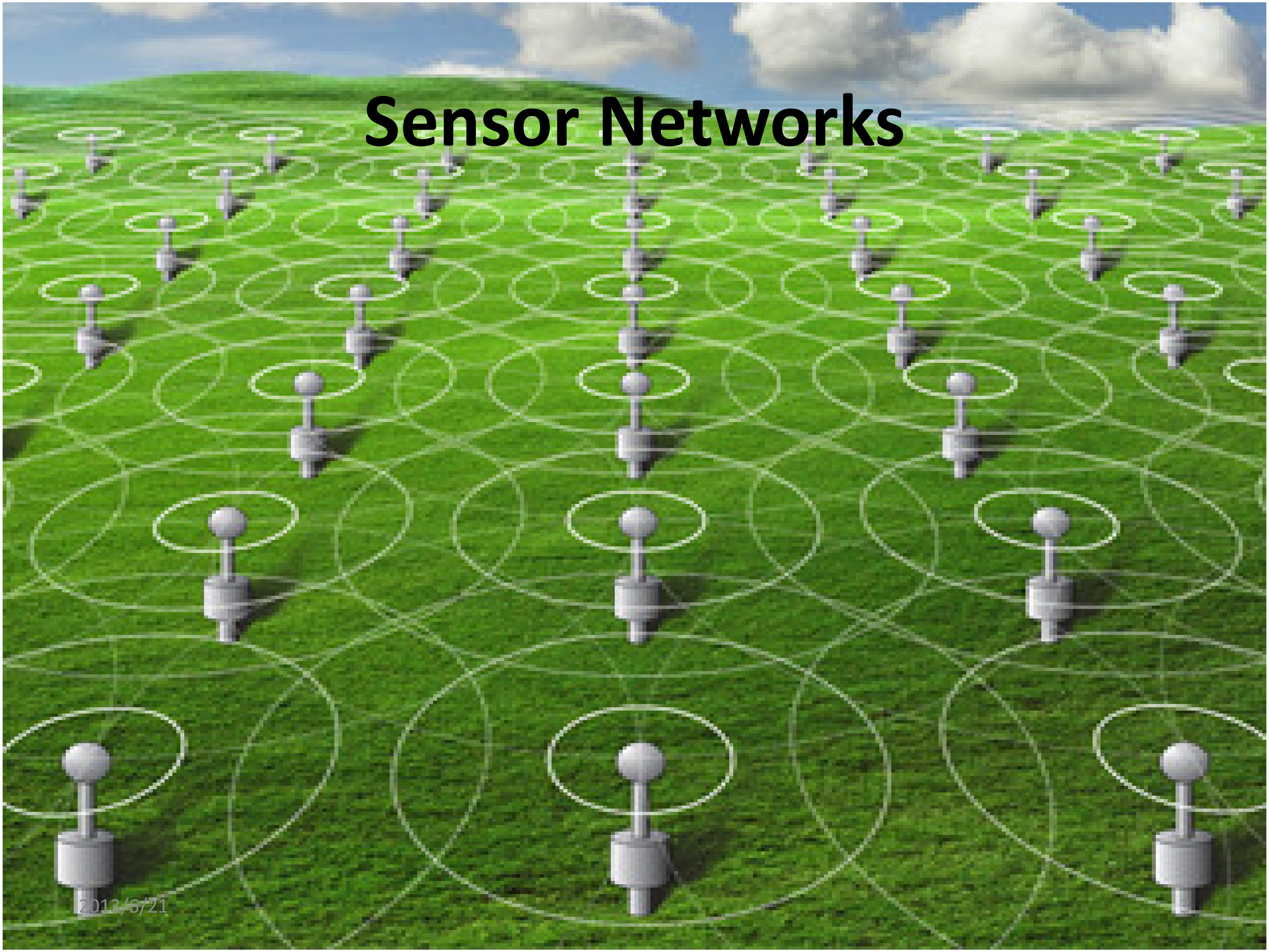
Jie Gao
Stony Brook University
CG Week 2012
Algorithms in the Field Workshop

A generic sensor node

- CPU.
- On-board flash **memory** or external memory
- Sensors: thermometer, camera, motion, light sensor, etc.
- **Wireless** radio.
- **Battery**.



Sensor Networks

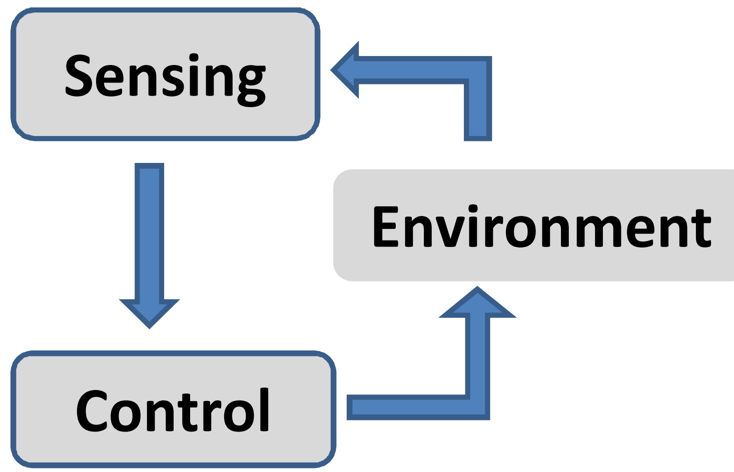


Applications

- Initially: **Scientific Monitoring**
  - High-resolution data collection and monitoring.
  - Deploy sensors in a remote region, collect data to a base station.
- Examples
  - Habitat monitoring of animal behaviors
  - Environment monitoring (redwoods, golden gate bridge, tunnels, etc)

Applications

- Later: **Cyber Physical Systems (CPS), Internet of Things (IoT)**
  – Real-time data acquisition.
  – Situation understanding.
  – Event response and control.



- Examples:
  – Health monitoring
  – Smart buildings; green buildings
  – SmartGrid

Applications

- Emerging: **Participatory Sensing**
  - Sensors on cellular phones (GPS, microphone, cameras…)
  - “Crowdsourcing”
  - Devices are available, charged & maintained.
- Applications
  - Social activities (foursquare)
  - Traffic monitoring (waze)

Where is the geometry?

1. Sensor locations
   – Points in the plane
   – Points on a terrain
   – Points in 3D
   – Mobile sensors

Where is the geometry?

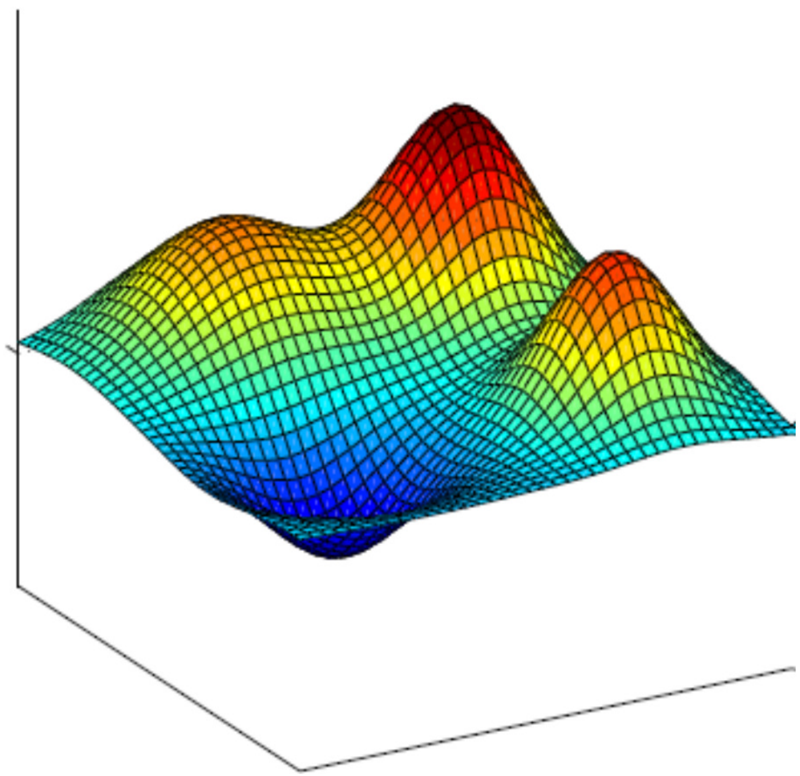
2. Sensor communication models
   - Short-range communication (for reducing energy consumption/interference)
   - E.g. Unit disk graph model (UDG)

Where is the geometry?

3. Sensing models
   - Isotropic: point, disk.
   - Non-isotropic: camera, laser.

Where is the geometry?

4. Sensor data
   - Spatial & temporal correlation in most physical phenomena



Model is important

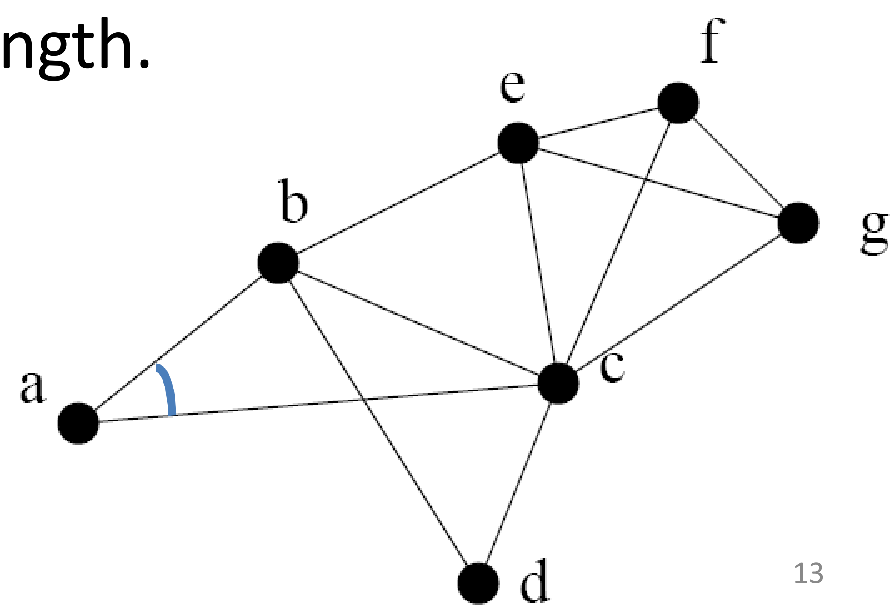
- Captures important properties
- Simple enough s.t. interesting results can be proved.
- Multiple interesting models & solutions.

Two case study

- Network localization
- Geometric routing

Network Localization

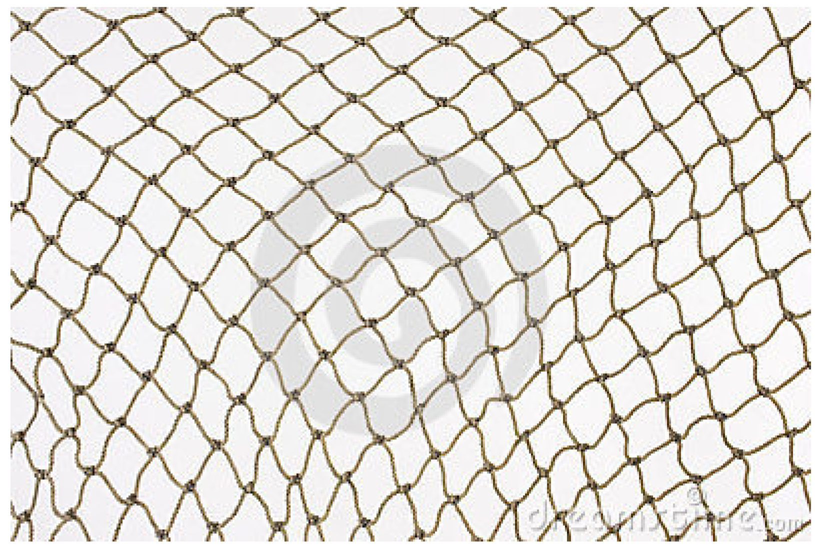
- Given a network of sensors, find their (absolute or relative) positions.
- Input:
  – Network connectivity.
  – (optional) noisy edge length.
  – (optional) noisy angles.



Unit disk graph realization

- Given a graph, can you realize it as a unit disk graph in R^2?

?



Unit disk graph realization

- Given a graph, can you realize it as a unit disk graph in R^2?
- NP-complete [Breu, Kirkpatrick'98]
- NP-hard to approx. within $\sqrt{1.5}$ [Kuhn, Moscibroda, Wattenhofer'04]
- NP-complete w. edge lengths [Aspnes, Goldenberg, Yang'04; Bâdoiu, Demaine, Hajiaghayi, Indyk'04]
- NP-complete w. angles [Bruck, Gao, Jiang'05]
- NP-complete w. noisy edge length & angles [Basu, Gao, Mitchell, Sabhnani'06]

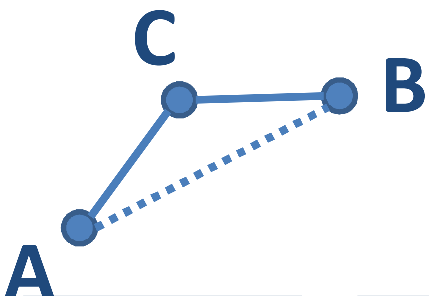
Unit disk graph realization

- Approximation?
- Known: $O(\log^{2.5} n \log \log n)$ on ratio of longest edge/ shortest non-edge [Kuhn, Moscibroda, Wattenhofer'04]
- **Open question**: $O(1)$-approximation?

Given a combinatorial UDG, embed all neighbors to be within distance 1, and all non-neighbors at least α apart, for a constant α.

Why is UDG realization so hard?

- Intuitively,



Not convex!

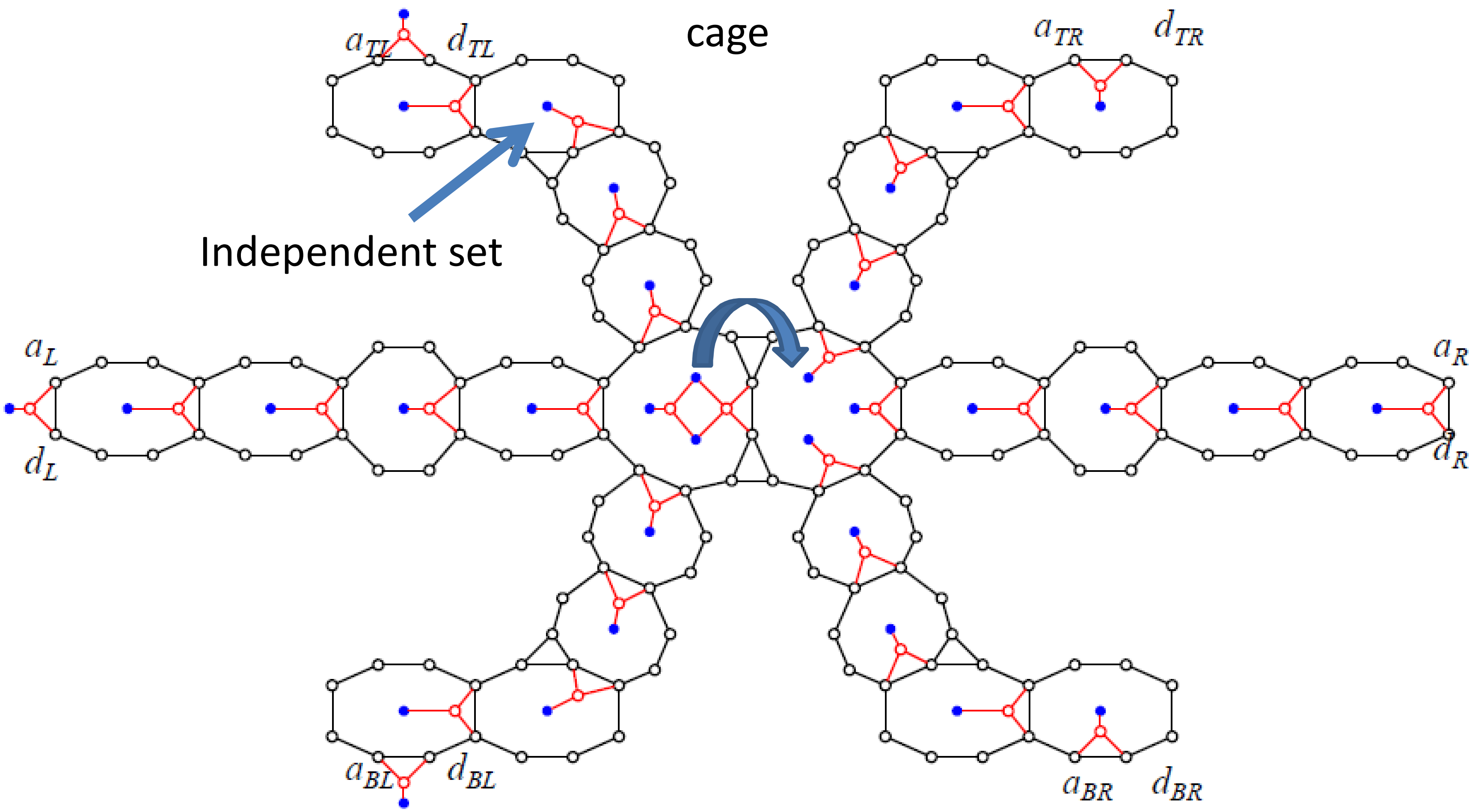
Packing disjoint disks rep. independent set!

AC, CB is an edge → |AC| ≤ 1, |BC| ≤ 1

AB is not an edge → |AB| >1

- Formally, all proofs use reduction from 3-SAT.

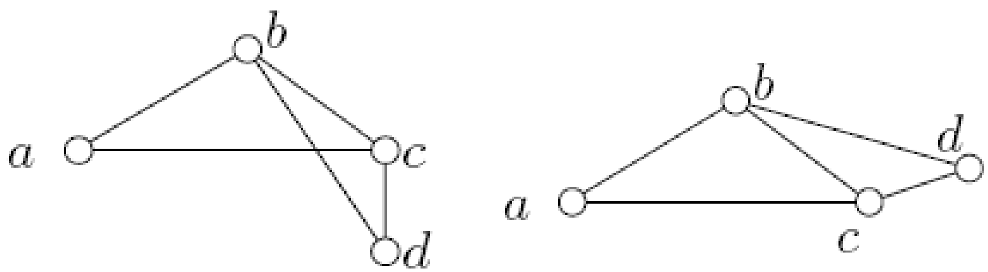
Truth setting gadget



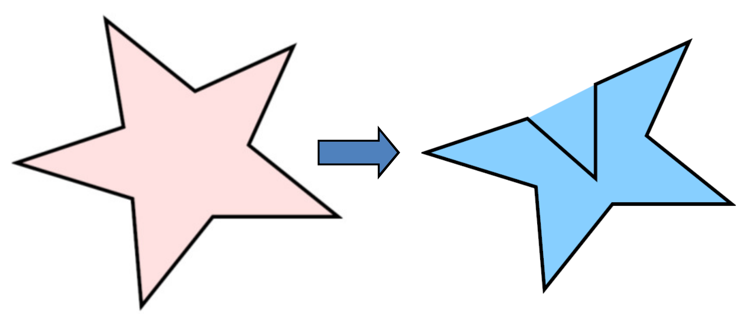
Unit disk graph recognition is NP-hard, Breu & Kirkpatrick, 1998.

2nd challenge: flip ambiguity

- Given two triangles with fixed edge length, one can flip one triangle relative to the other.

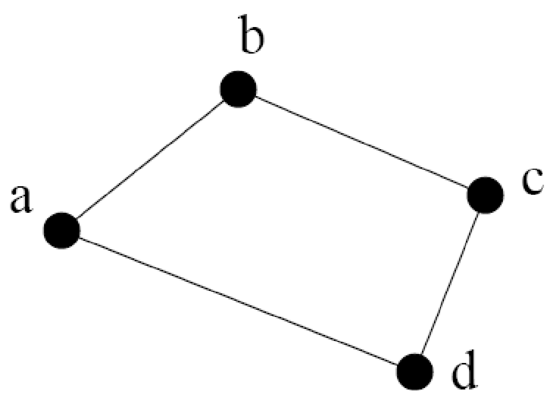


- Incorrect global flip.

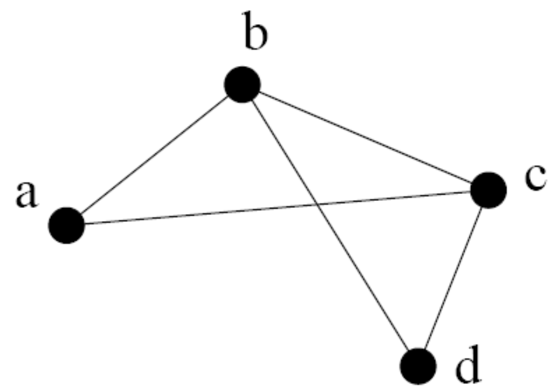


Rigidity and global rigidity

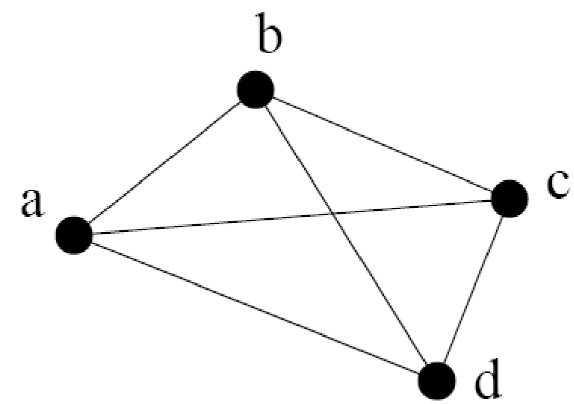
- **Rigidity**: no continuous deformation
- **Global rigidity:** unique embedding in R^2



Not rigid



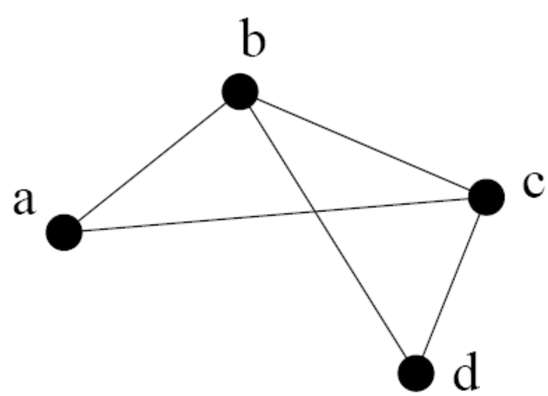
Rigid, not globally rigid

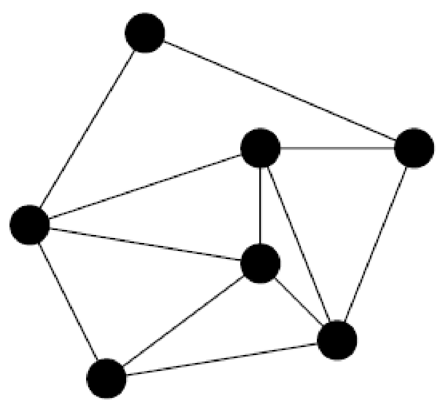


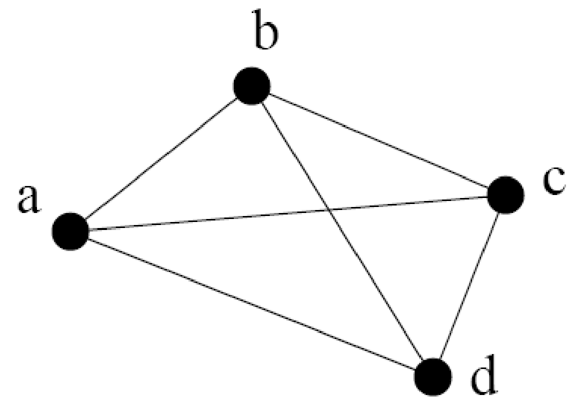
Globally rigid

2D rigidity is well understood!

- A graph is rigid in 2D if and only if it contains a **Laman graph** on its vertices.
- A Laman graph has n vertices, 2n-3 edges and any subset of k vertices spans at most 2k-3 edges.

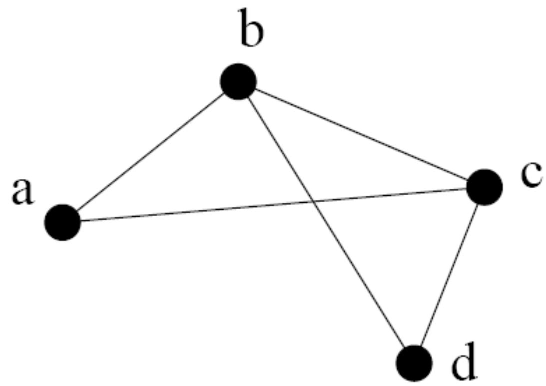


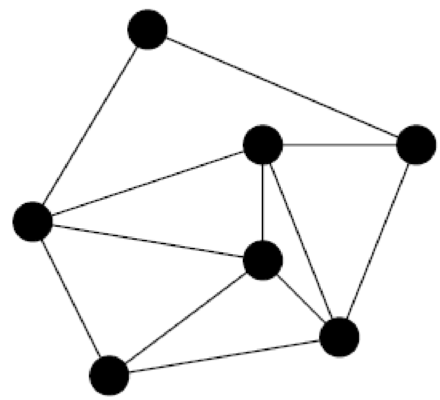


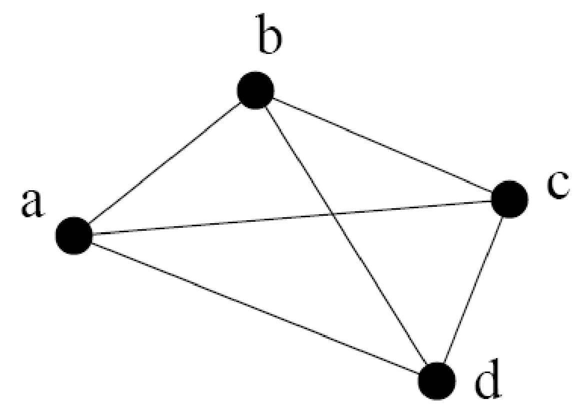


Global rigidity

- A graph is globally rigid in 2D iff it is 3-connected and redundantly rigid (rigid upon the removal of an edge).







Rigid, not globally rigid

Globally rigid

[Berg, Jordan 2003] [Hendrickson 1992]

Recognizing rigidity is easy

- **Pebble game**: test whether a graph is generically rigid in time O(nm), n=|V|, m=|E|.
- Testing global rigidity?
  – Test redundant rigidity
  – Test 3-connectivity.

D. J. Jacobs and B. Hendrickson, An Algorithm for two dimensional rigidity percolation: The pebble game. J. Comput. Phys., 137:346-365, 1997.

But finding an embedding is hard!

- Given a graph with edge lengths specified, finding a valid graph realization in R^d for a **fixed** dimension *d* is NP-complete.
- Rank constraint is non-convex.
- 1st idea: embed in $\mathbf{R^n}$ and project down to R^2 – multi-dimensional scaling.
- 2nd idea: poly time algorithm for **special families of graphs.**

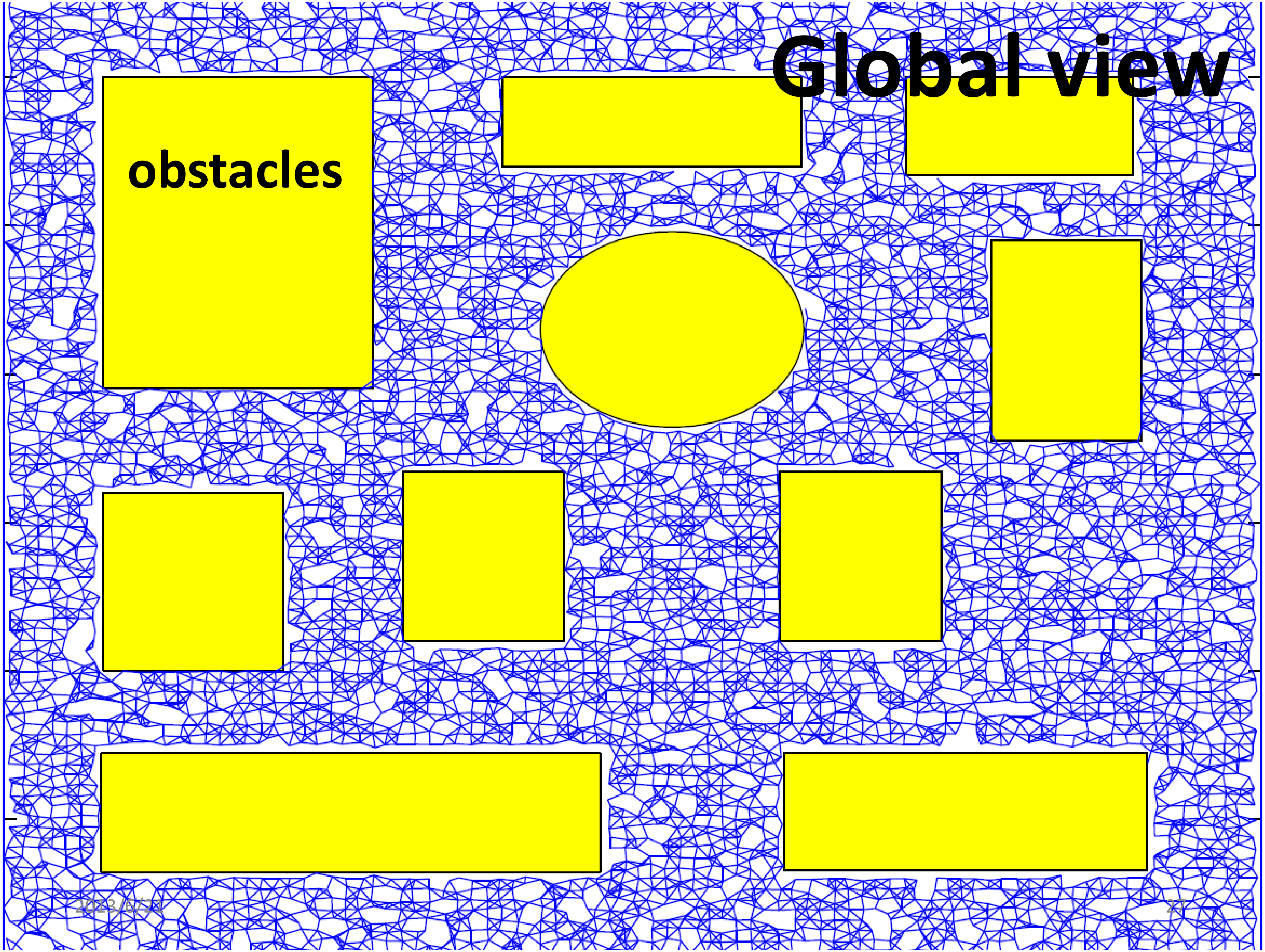
What about dense graphs?

A detour

- Look at a different model of the network.

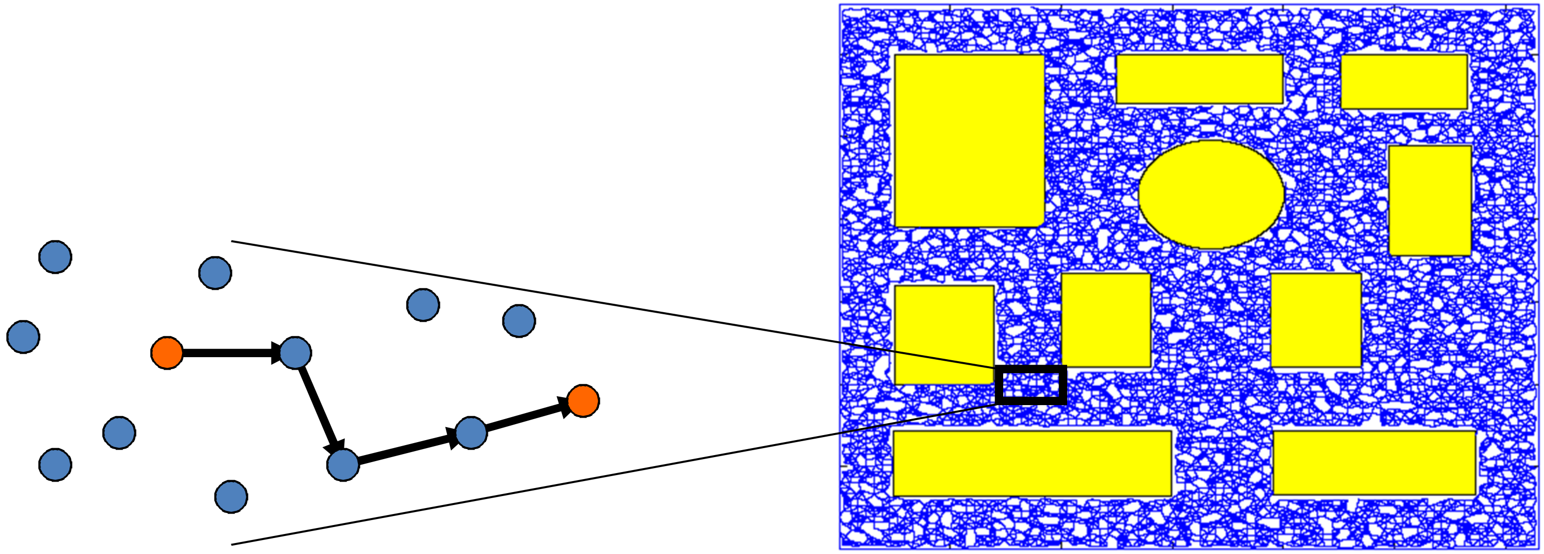
Local view

Global view



Large-scale sensor field

- large-scale, dense deployment.
  - Sufficient sensor coverage.
- w/ holes/obstacles.
- Prominent feature: **shape** of the network.

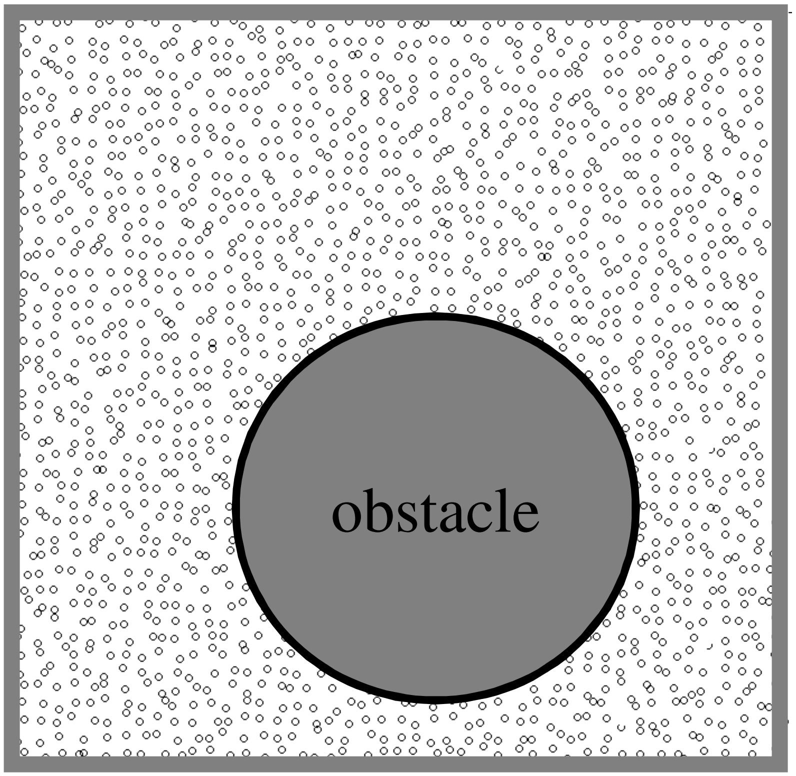


Graph abstraction of sensor networks

- The “graph” view of a sensor network depends very much on the network connectivity, which is <u>not stable</u>: links come and go, nodes fail…
- Dynamic graph problems, in general, are difficult problems.

Geometric View of Sensor Networks

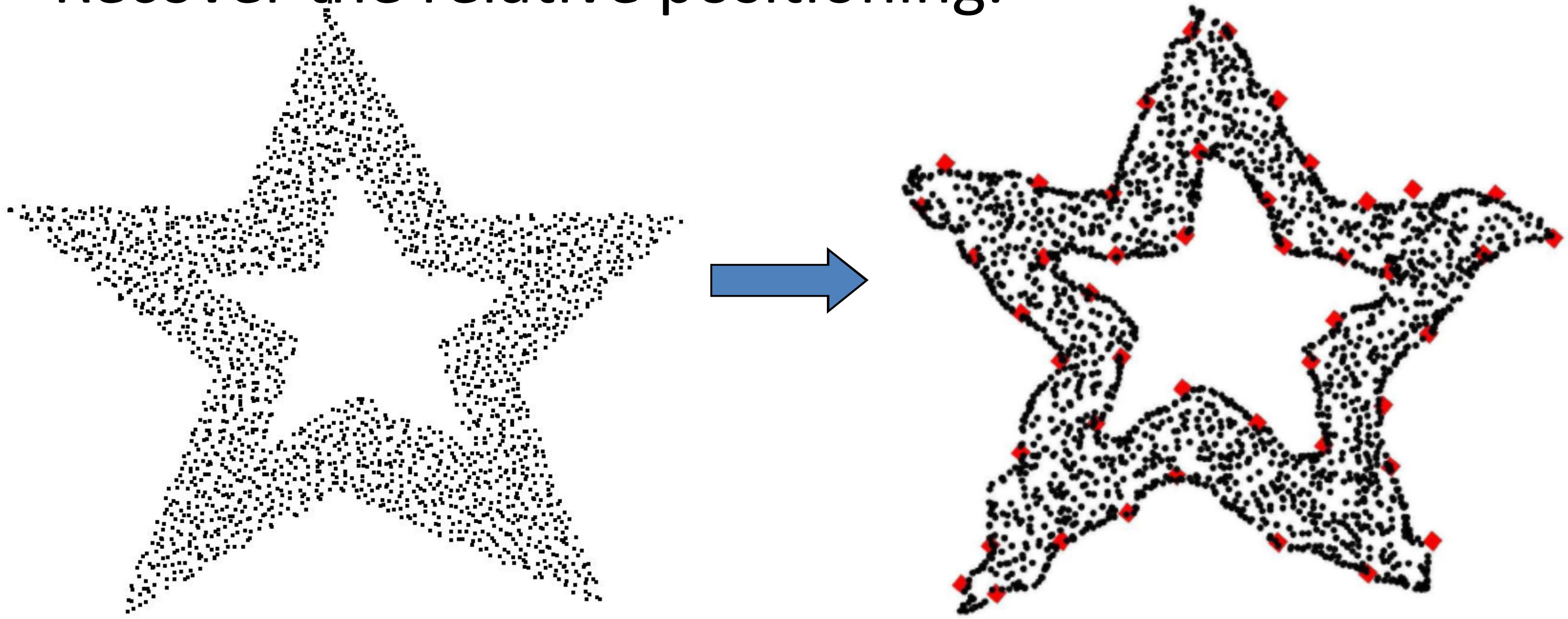
- The global geometry/ topology is stable
- We know how to handle shapes



Network Localization

Surface reconstruction?

- Large, dense sensor field with complex geometry.
- Nearby nodes are able to communicate.
- Use connectivity information only.
- Recover the relative positioning.

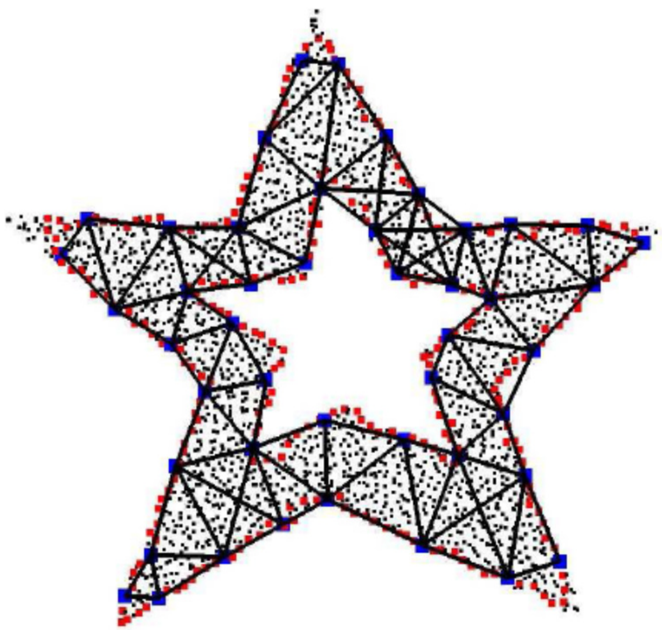


The algorithm

- Select landmarks on the boundary.

Landmark Voronoi diagram: Nodes identify closest landmark under network distance

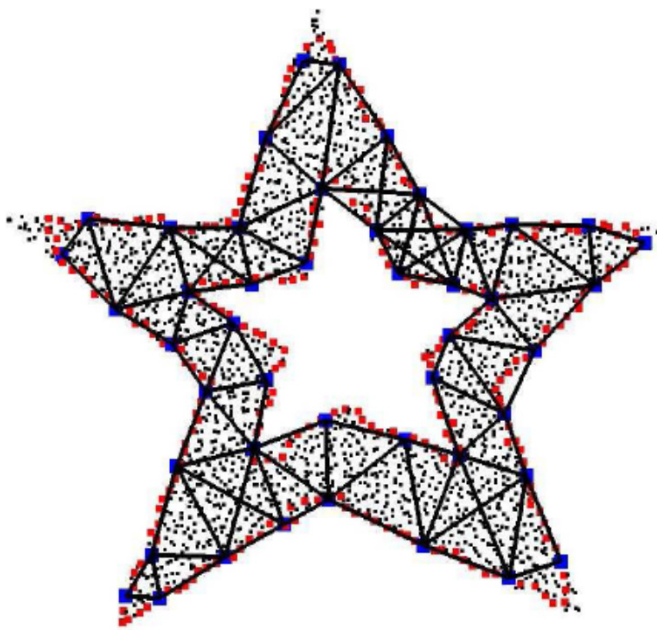
Combinatorial Delaunay complex: Neighboring landmarks connected by an edge

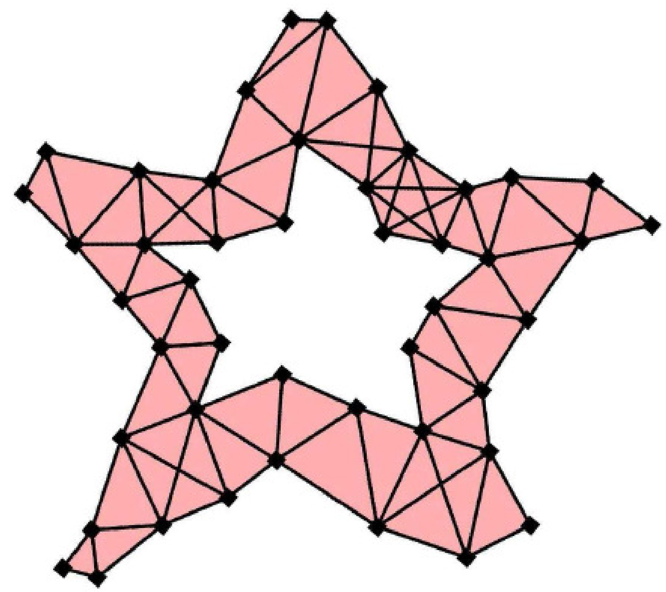


[Lederer, Wang, Gao, 2008].

The algorithm, cont

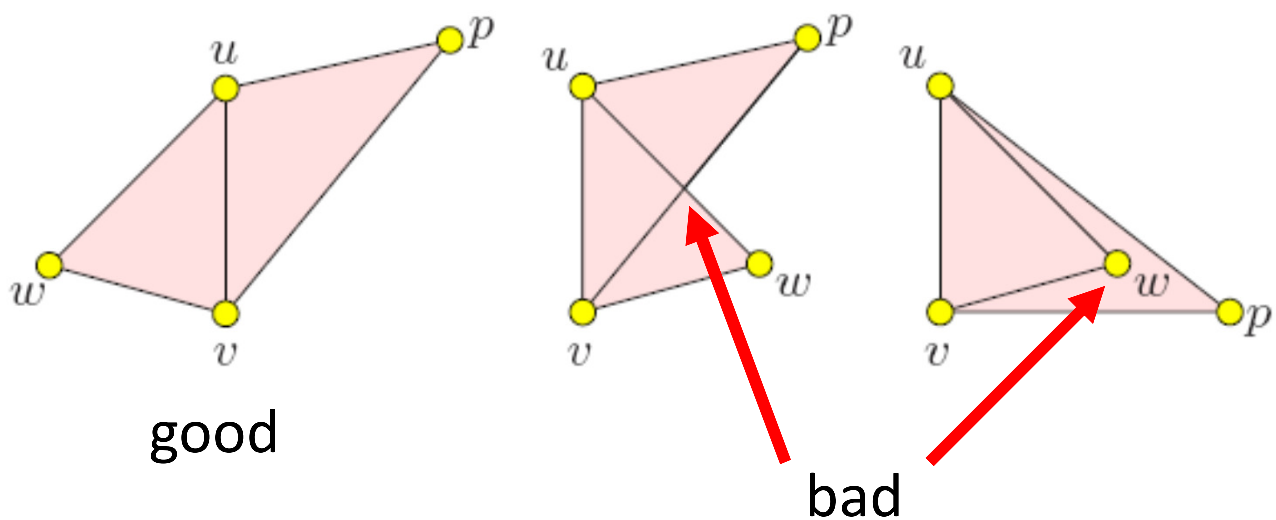
- Embed the combinatorial Delaunay complex
  - Edge length = network hop count
  - Glue adjacent triangles side by side.





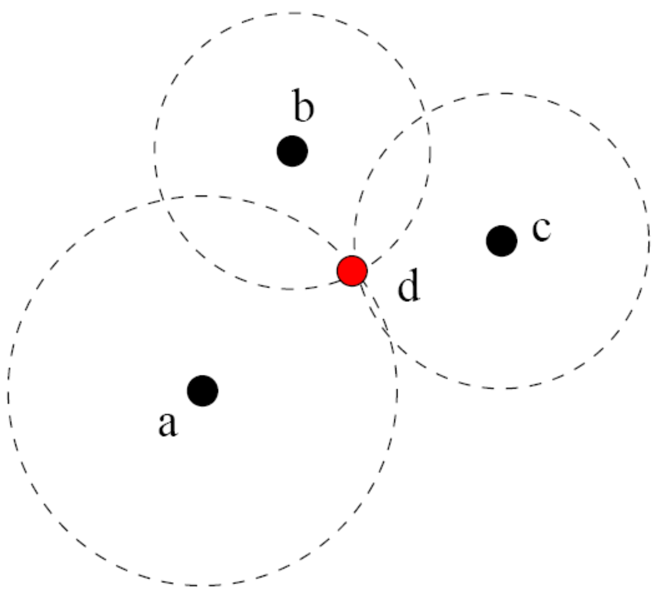
No flip ambiguity for Delaunay triangles

- Recall: we want to embed as a simplicial complex.
  - Intersection of two simplices is empty or is a common face.



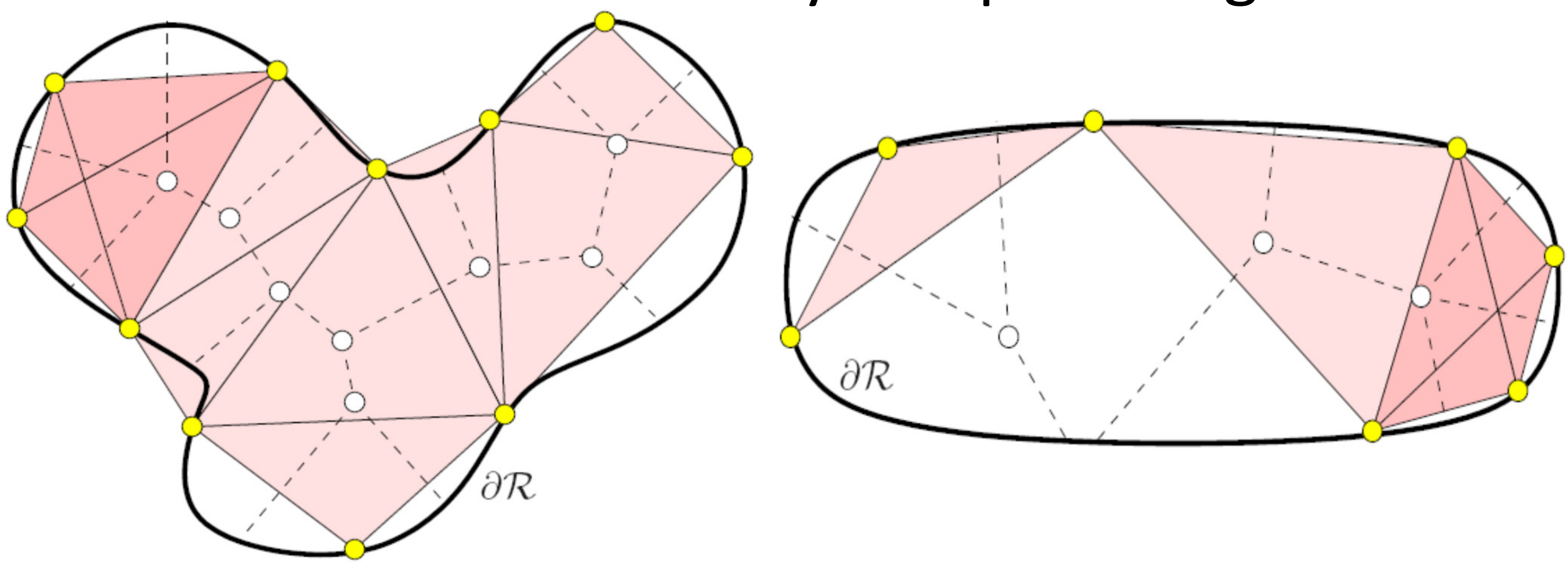
Algorithm, cont

- Embed the rest of the nodes.
  - Each node embeds itself with hop-count distances to nearby 3 landmarks.



Are we done?

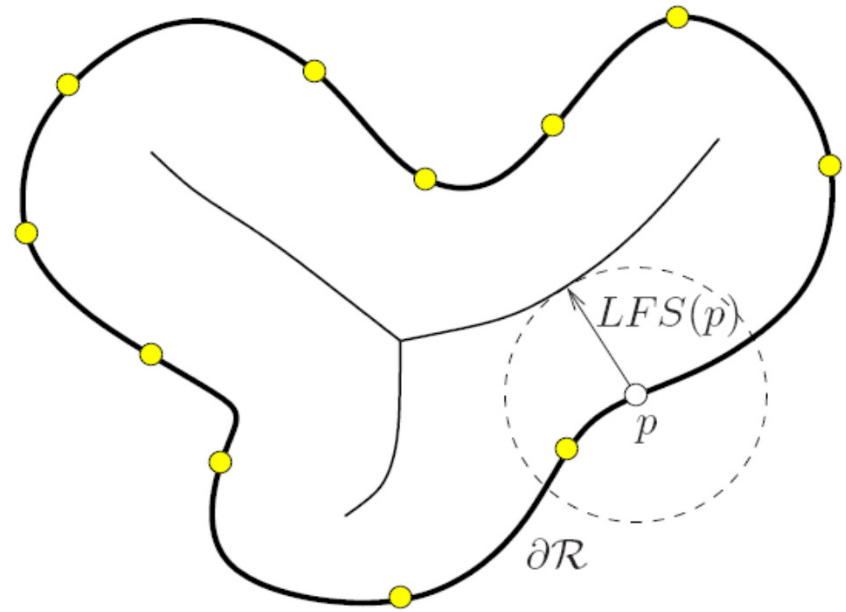
- How to select landmarks?
- Make sure the Delaunay complex is rigid.



Landmark selection condition

- If the Voronoi diagram (Voronoi edges and vertices) is connected in R, then the Delaunay graph is rigid.
- Sampling criterion:

Each boundary point p has a landmark within LFS(p).

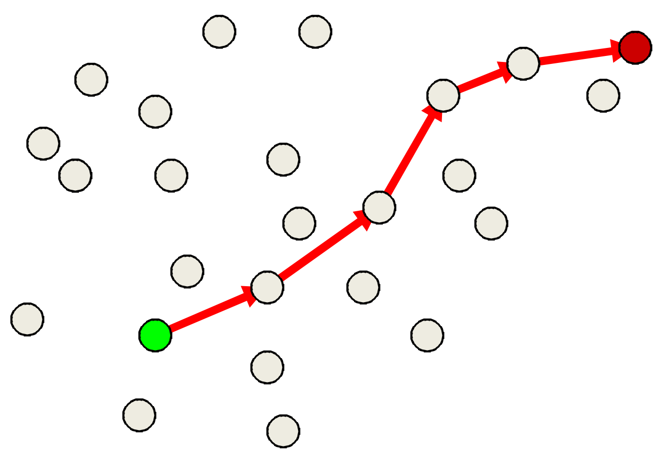


Two case study

- Network localization
- **Geometric routing**

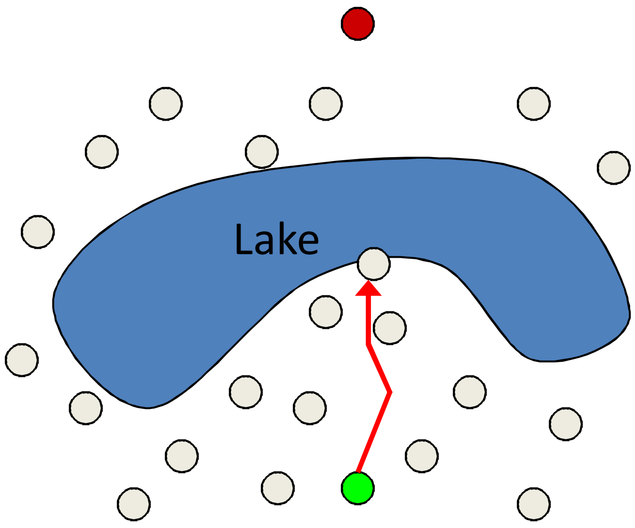
Greedy Routing in Sensor Networks

- Assign coordinates to nodes
- Message moves to neighbor closest to destination
- Simple, compact, scalable



Greedy routing may get stuck

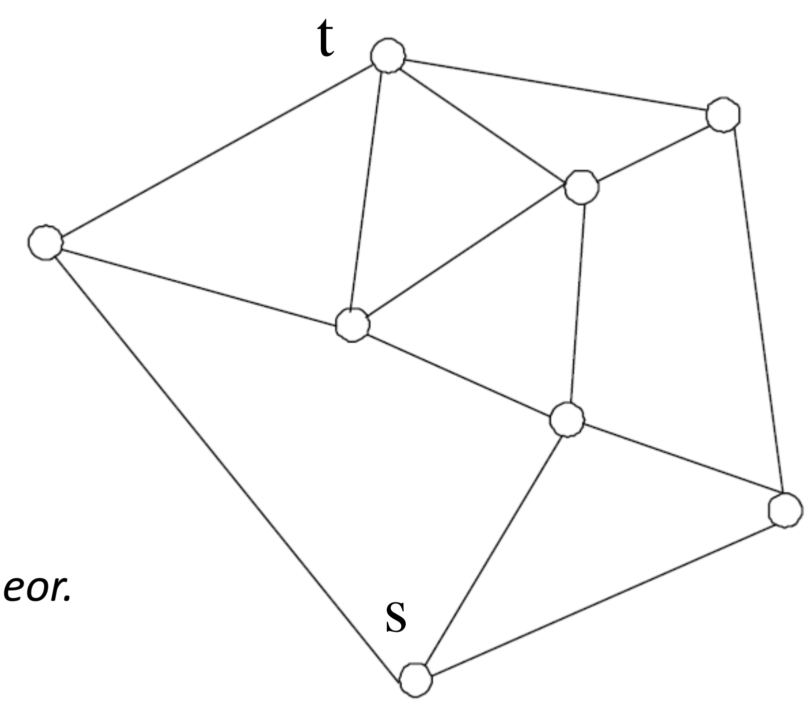
- Can get stuck



- Find a different embedding so that greedy routing does not get stuck!

Greedy embedding

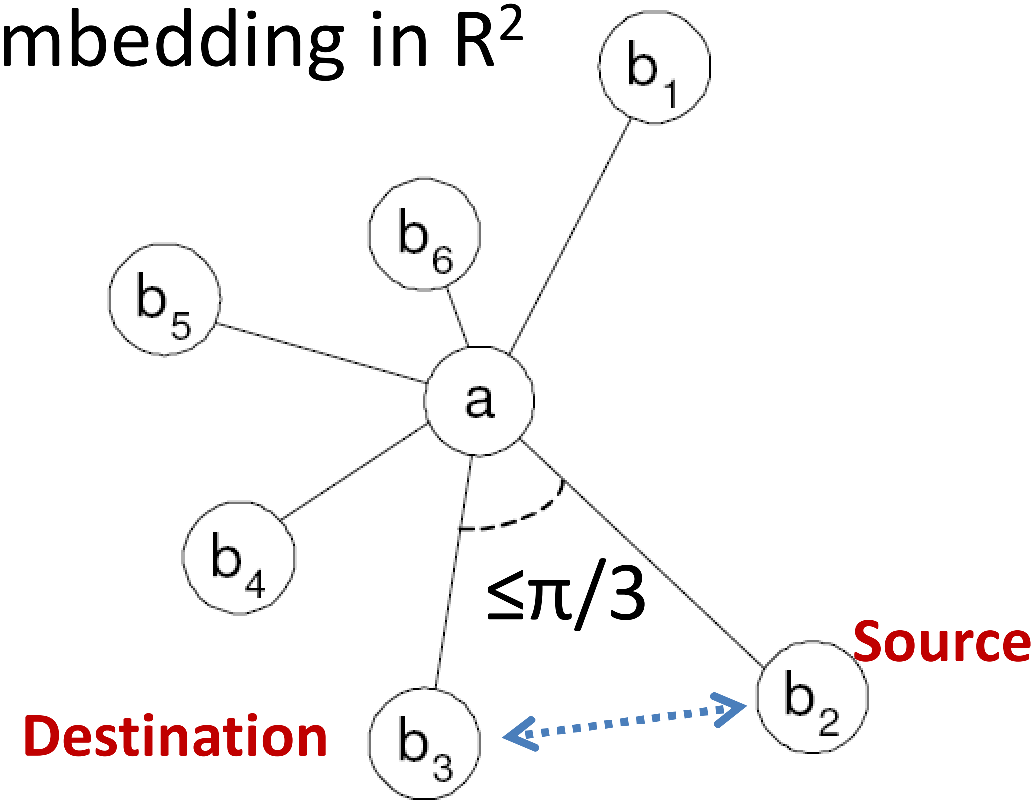
- Given a graph G, find an embedding of the vertices in R^d, s.t. for each pair of nodes s, t, there is a neighbor of s closer to t than s itself.
- Q: how to compute?



C. H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. *Theor. Comput. Sci.*, 344(1):3–14, 2005.

Greedy embedding does not always exist

- A star with ≥ 6 leaves does not have a greedy embedding in R^2

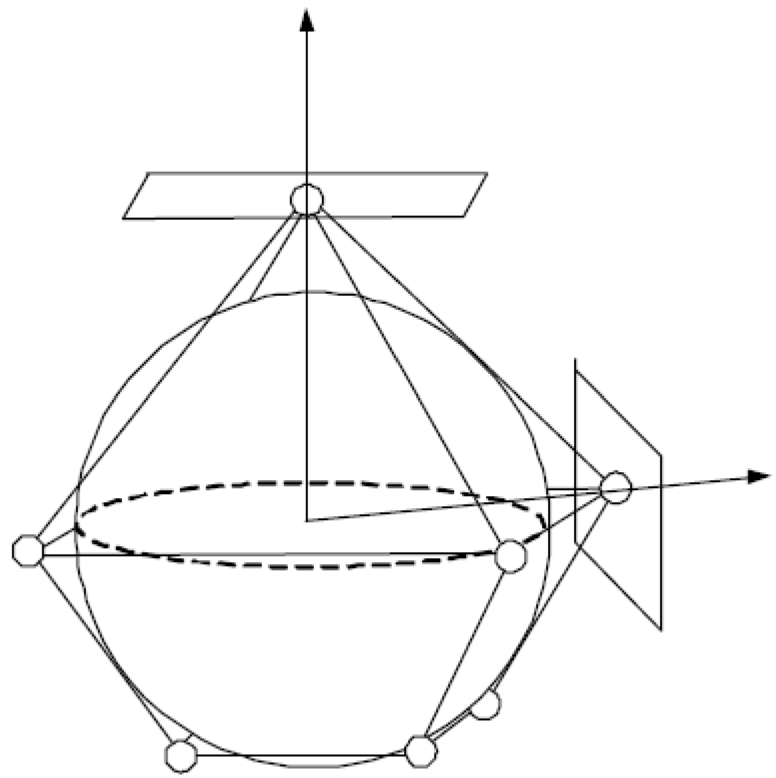


Not the longest distance in Δab_2b_3

Conjecture [Papadimitriou, Ratajczak]

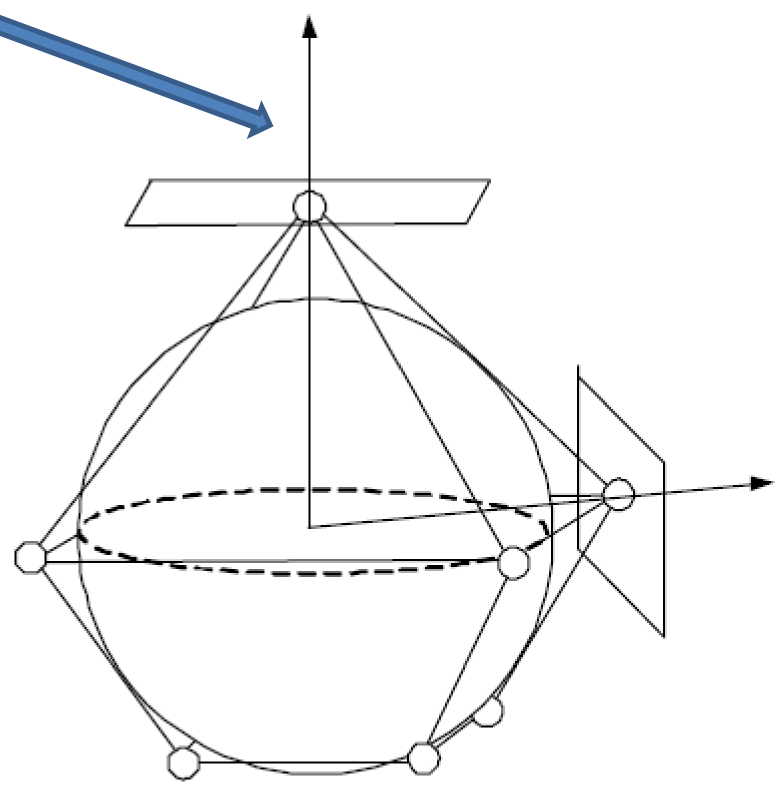
- Any **planar 3-connected** graph has a greedy embedding R^2.

- Any 3-connected planar graph is the edge graph of a **3D convex polytope**, with edges tangent to a sphere. [Steinitz 1922].



Polyhedral routing

Theorem: Greedy routing with the distance function $d(u, v) = -\mathbf{e}(u)\cdot\mathbf{e}(v)$ guarantees delivery.



1. Distance is a linear function ➔ single global minimum.
2. For destination v, d(u, v) is minimum if u=v.
3. No stuck point due to convexity.

Progress on the conjecture

- Dhandapani proved that any triangulation admits a greedy embedding (SODA’08).
- Leighton and Moitra proved the conjecture (FOCS’08).
- Independently, Angelini et al. also proved it (Graph Drawing’08).

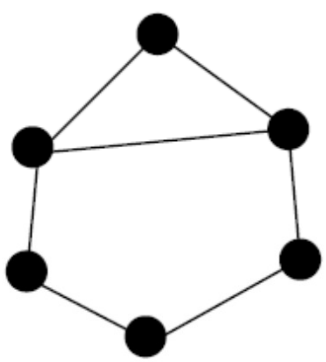
Leighton and Moitra's Proof

- All 3-connected planar graphs contain a spanning Christmas Cactus graph.
- All Christmas Cactus graphs admit a greedy embedding in the plane.

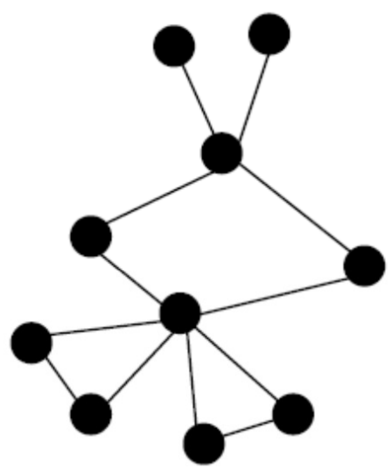
A Christmas Cactus

Christmas cactus graph

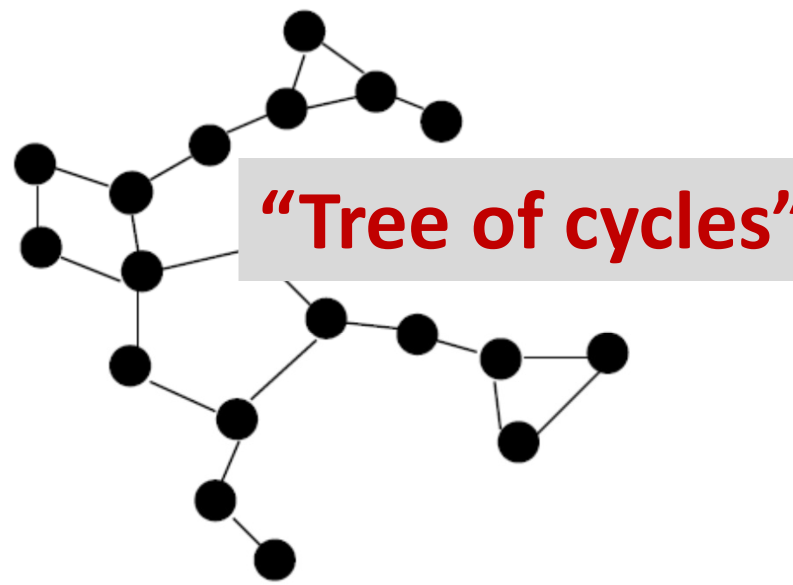
- A cactus graph is connected, each edge is in at most one simple cycle.
- A Christmas Cactus graph is a cactus graph for which the removal of any node disconnects it into at most 2 pieces.



not cactus

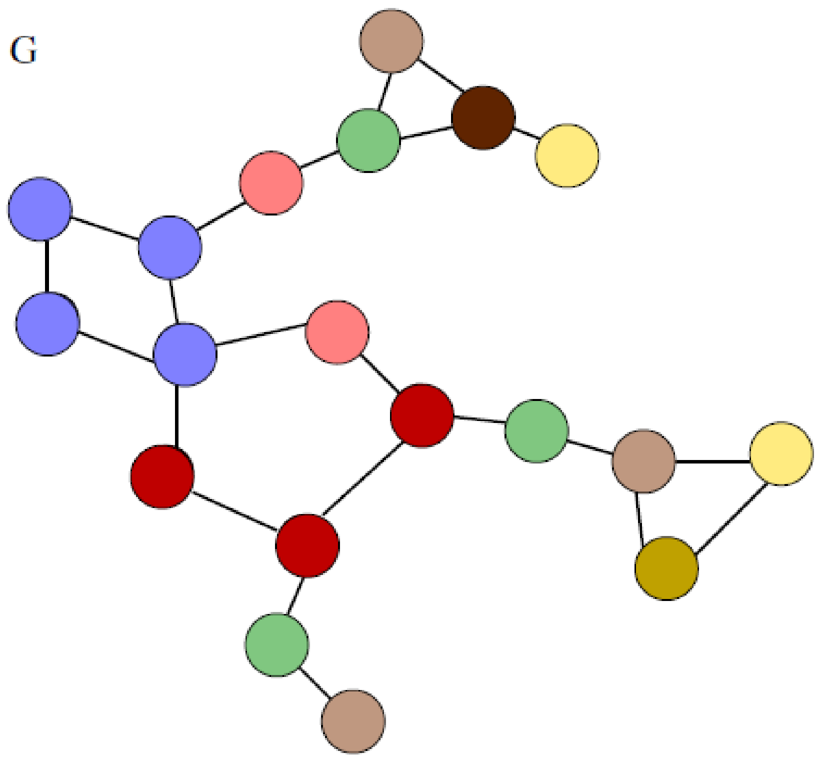


cactus, not Christams cactus

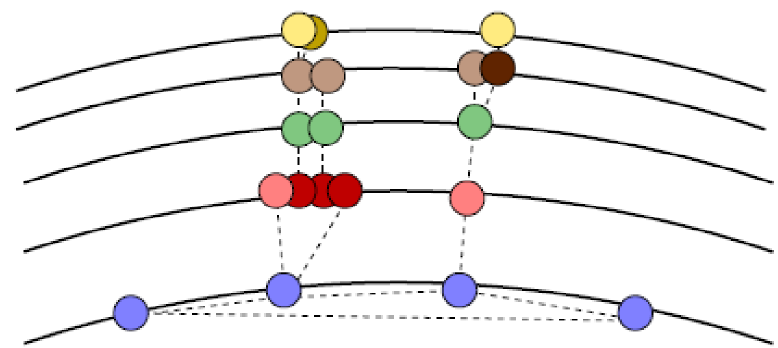


Christmas cactus

How does the embedding look like?



Leighton and Moitra



- Requires high resolution!
- $\Omega(n \log n)$ bit size coordinates! [Eppstein, Goodrich'08]

Succinct greedy drawing

- Goal: o(n log n) bit size coordinates.
- But this is in general impossible. [Battista, Frati, Angelini, 2009].
- Notice: no need to store exact coordinates, only need comparison.
- Goldrich and Strash show compressed representation of "Christmas cactus embedding" using O(log n) size storage.

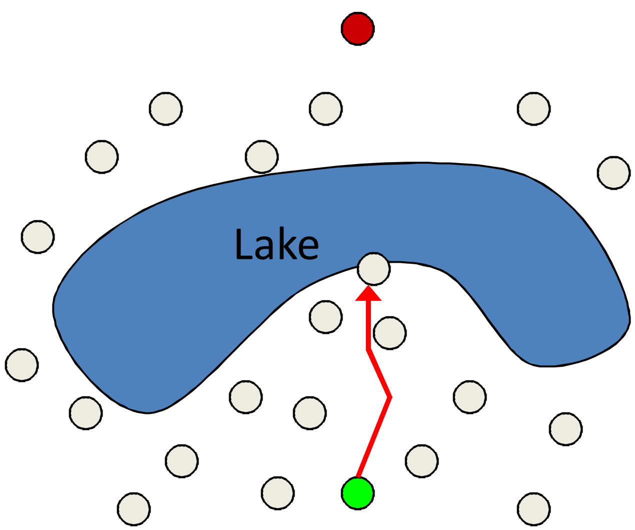
Greedy drawing

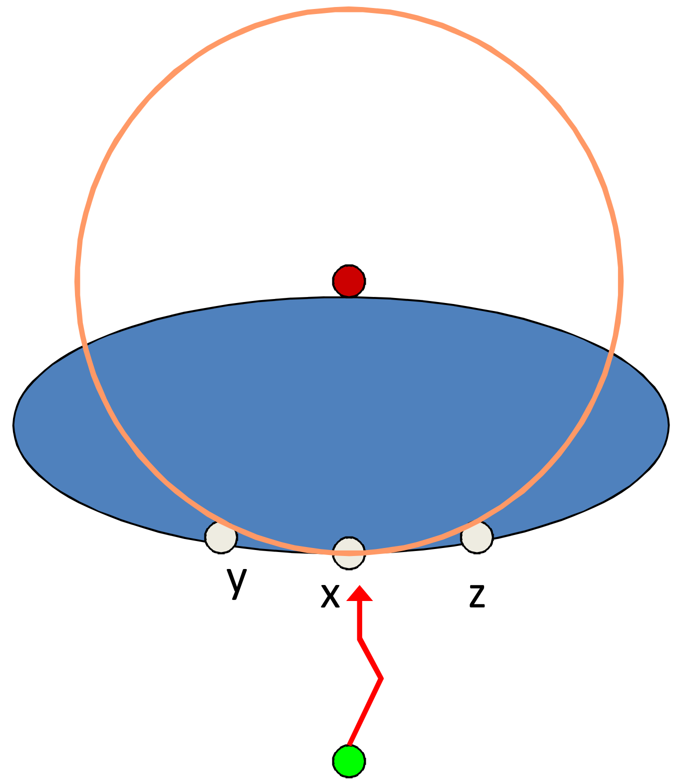
- Very nice theory
- In practice: links go up and down.
- Maintaining the greedy drawing is non-trivial.

- Switch to the geometric view.

Greedy routing get stuck at “holes”

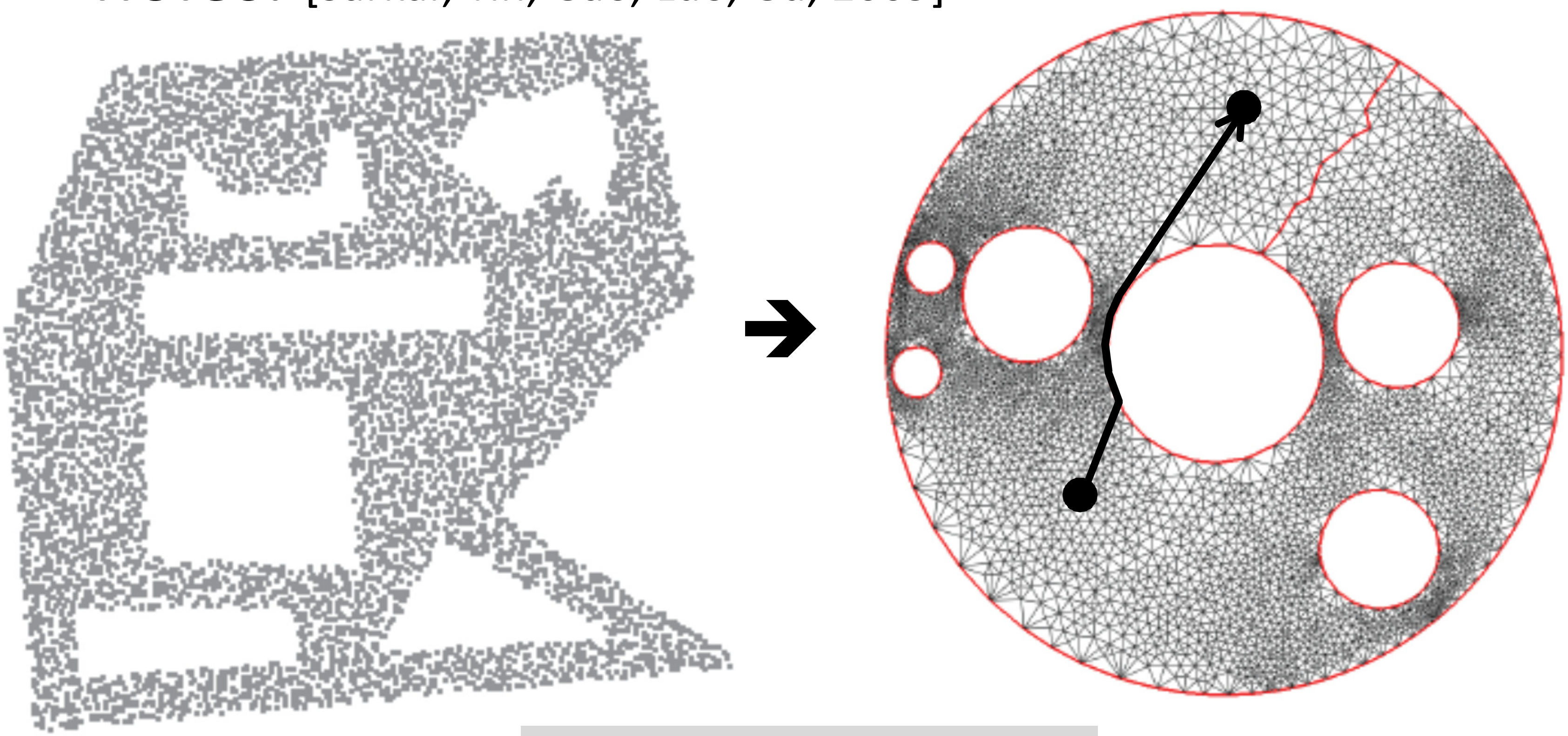
- Can get stuck





Deform the holes to be circular

- Greedy routing does not get stuck at circular holes. [Sarkar, Yin, Gao, Luo, Gu, 2009]



Conformal map!

Conformal map for greedy routing

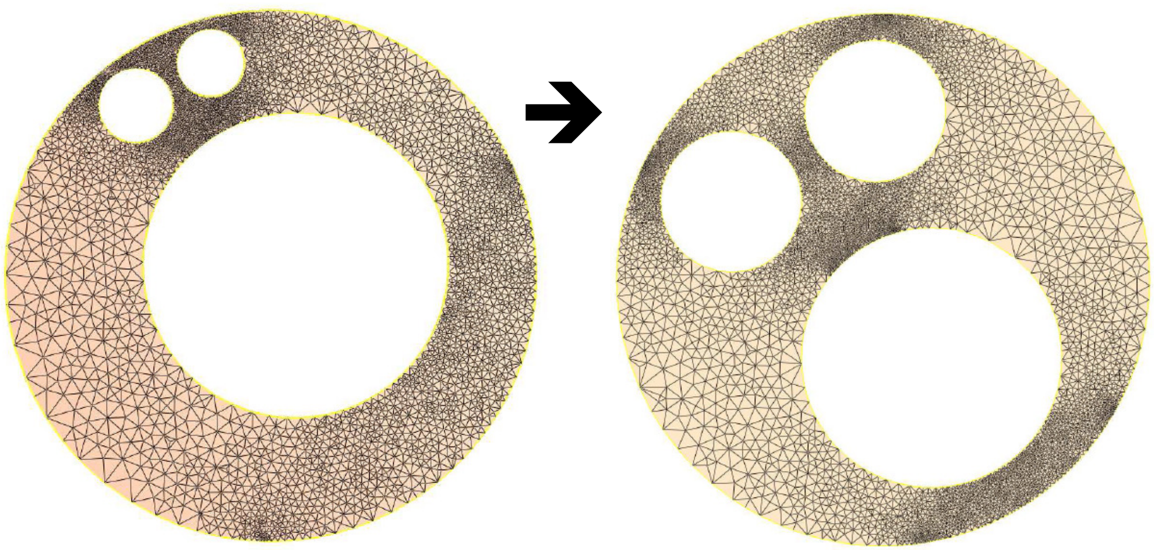
- Curvature flow (Ricci flow) is a natural **distributed** algorithm.
- Deform a complex shape to a simple one, making it easy to explore the **space of paths**.
  - Multi-path routing
  - Recover from link failure
  - Find routes of different homotopy types

Möbius Transform

- Möbius transform
  - Conformal: maps circles to circles

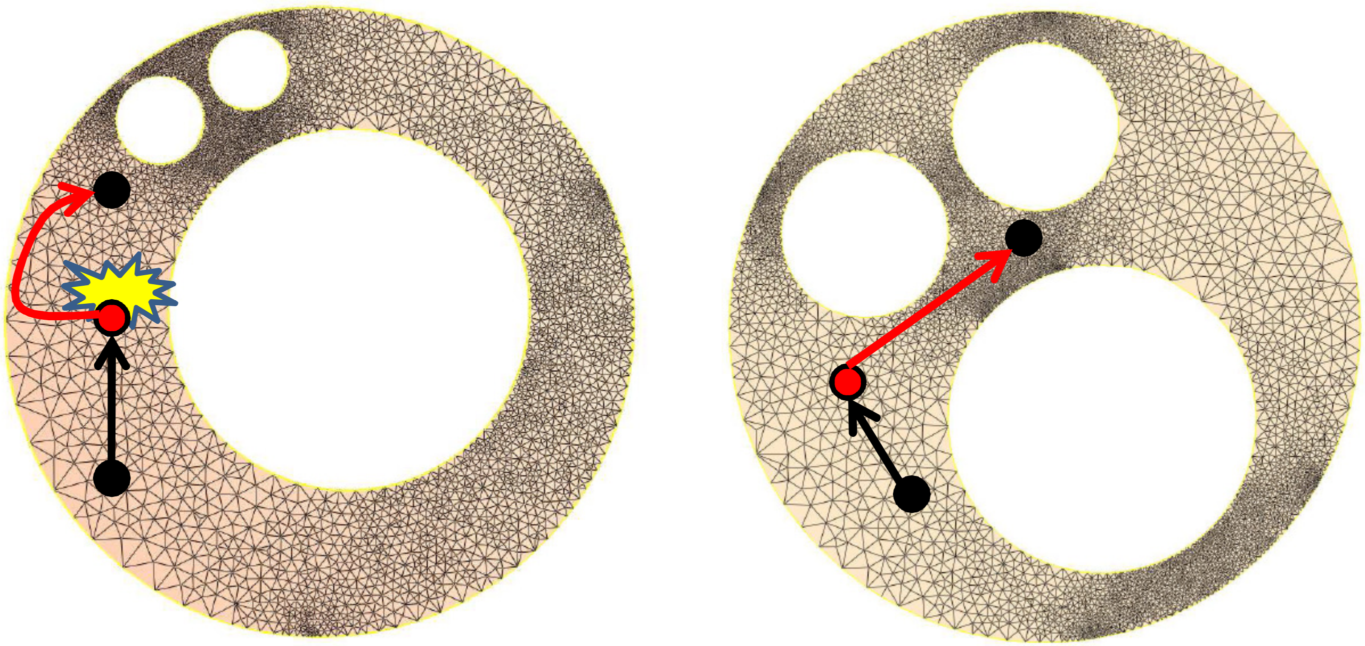
$$f(z) = \frac{az + b}{cz + d}$$

a, b, c, d are 4 complex numbers, $ad \neq bc$



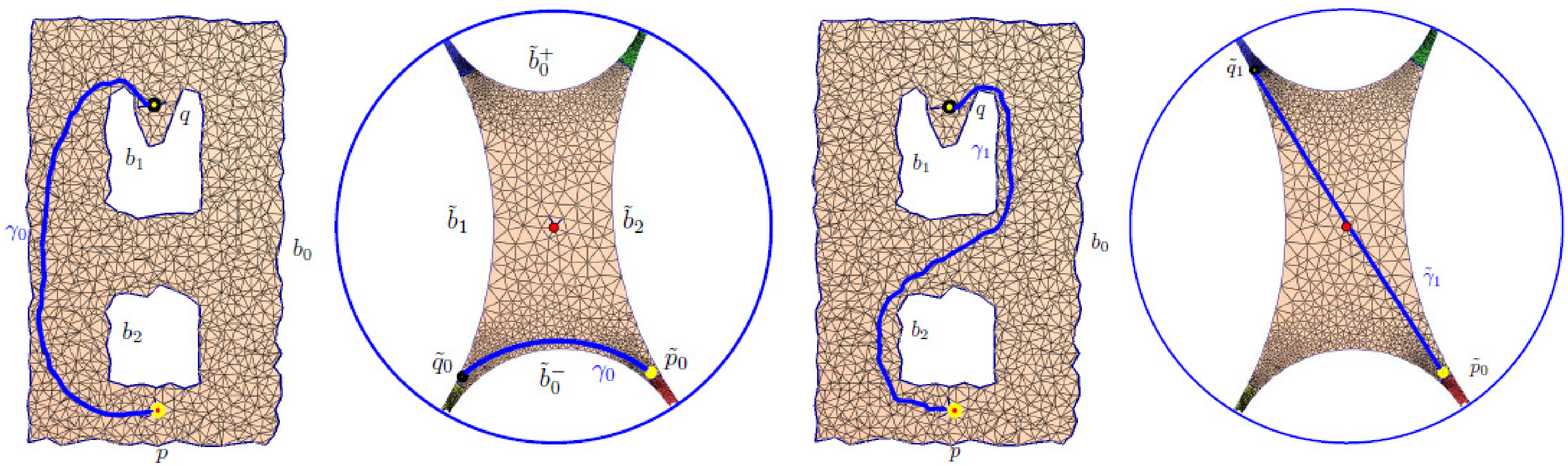
Recover from link failure

- Apply a Möbius transformation & route in an alternative embedding. [Jiang, Ban, Goswami, Zeng, Gao, Gu, 2011]



Embed into hyperbolic plane

- Greedy routing on the **universal covering space** finds paths of different **homotopy** types. [Zeng, Sarkar, Luo, Gu, Gao, 2010]

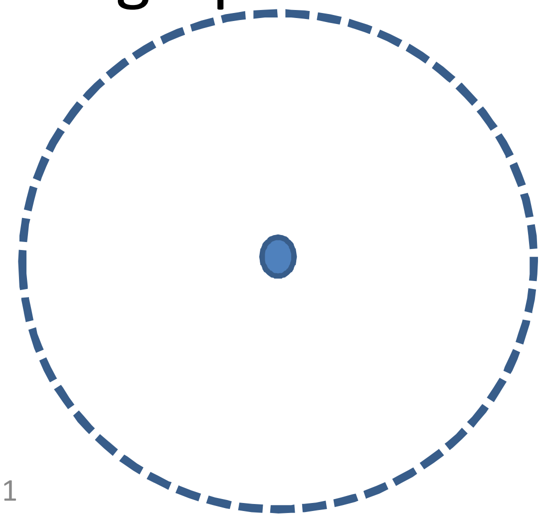


Conclusion

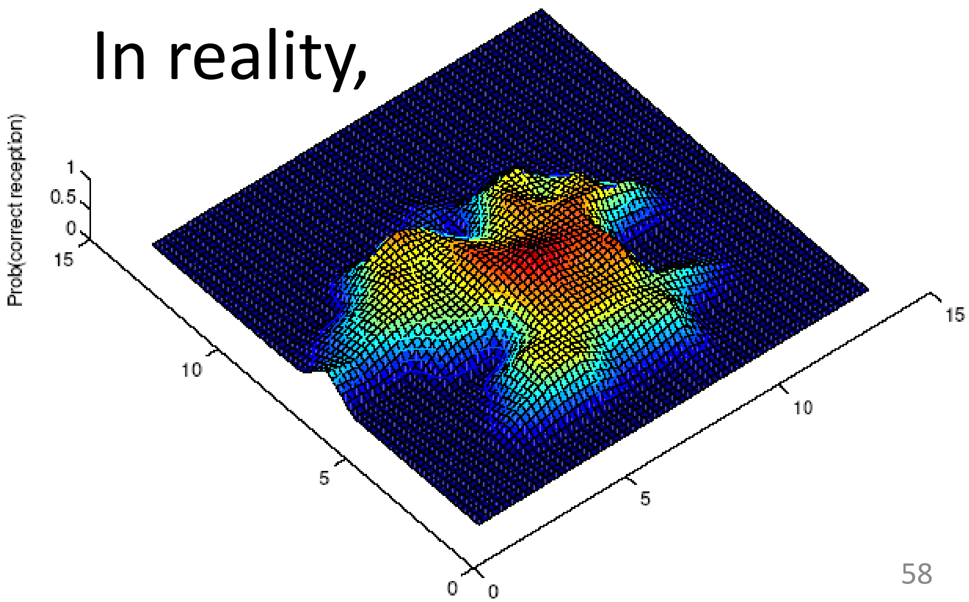
- Many nice geometric problems
- Modeling the problem is important

"All models are wrong, but some are useful."
[George E. P. Box]

Unit disk graph model



In reality,



Questions and Comments

- http://www.cs.sunysb.edu/~jgao
- jgao@cs.sunysb.edu

- Jie Gao and Leo Guibas, Geometric Algorithms for Sensor Networks, Philosophical Transactions of the Royal Society A, 2012.